****

**《Python程序设计》**

**课程设计报告**

**（2022年秋季学期）**

**学 院 名 称： 计算机科学与技术学院**

**设 计 题 目： 五子棋小游戏**

**学 生 姓 名 ： 安玉坤**

**学 号： 202103010056**

**专 业 班 级： 计科21-2**

**开 课 教 师： 姚 艳**

**完 成 时 间： 2022年12月23日**

1. **设计选题**

五子棋小游戏（Gobang）

1. **课题简介**

基本功能：人人对战，人机对战，悔棋，重开，再来一局，基本的五子棋功能,胜利后会提示具体哪五个棋子五子连珠

采用技术：pygame的画图，map函数，lambda表达式，鼠标，键盘事件实时监听，easyui编写交互界面

新意：整个棋盘是一个15\*15的二维数组，没有落子是0，落黑子是1，落白子是2，初始化使用numpy.zeros([15, 15], dtype=int),numpy可以提高处理数组的效率，判断输赢的时候采用了两种方法，一种是将棋子数组转化成字符串，判断“11111”或者“22222”是否在其中，另一种是循环遍历，也可以使用沿对角线遍历数组的方法，下棋本质是入栈，悔棋就是出栈

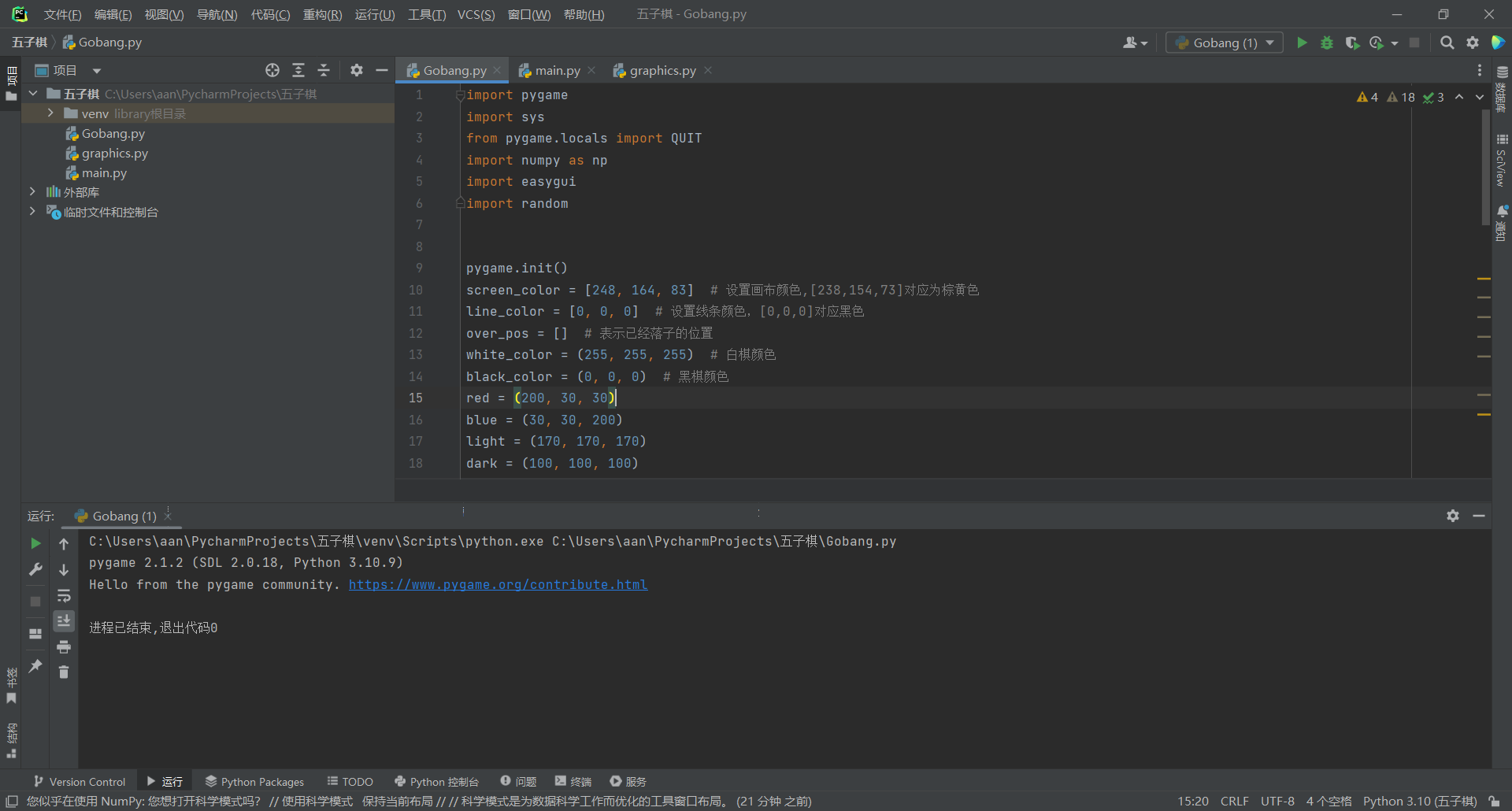
**三、运行环境**

开发环境：pycharm(python3.0)

引用的包：pygame,sys,numpy,easygui,random

**四、具体实现**

**1. 代码结构说明**



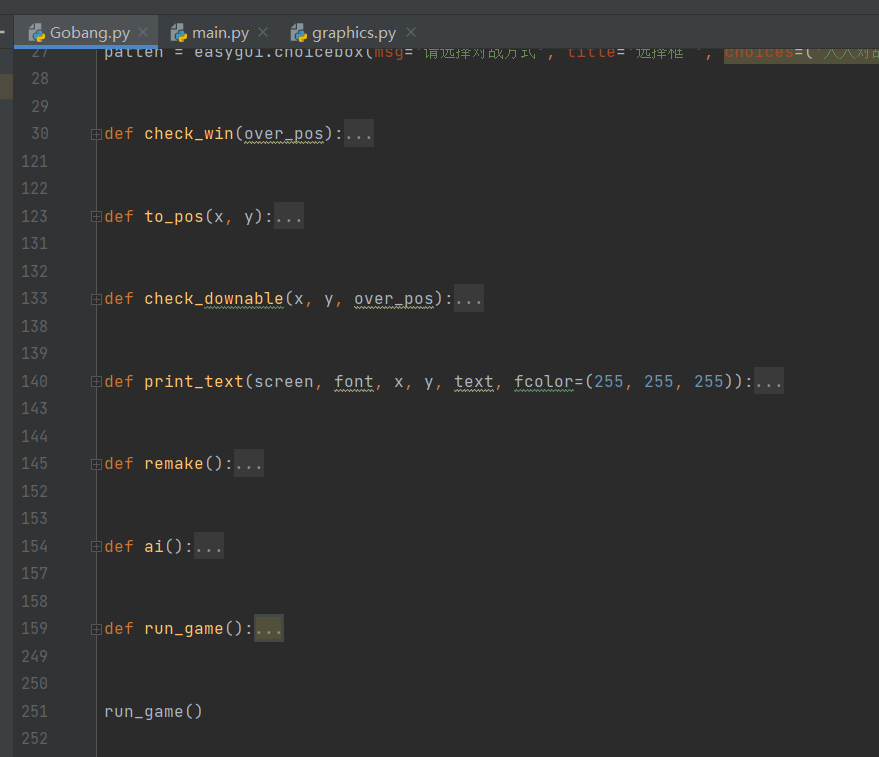


图1 系统总体框架与结构示意图

下面分别介绍各个部分的作用及功能：

check\_win（）判断是否有玩家胜利；

to\_position（） 将鼠标点击的坐标转化成五子棋的落点坐标；

check\_downable（）检查当前位置是否可以落子；

ai() ai下棋；

Run\_game() 运行游戏；

**2. 核心功能实现（给出核心功能的源代码。）**

实时监听鼠标、键盘事件，判断输赢

while True:

for event in pygame.event.get(): # 获取事件，如果鼠标点击右上角关闭按钮，关闭

if event.type == QUIT:

sys.exit()

if event.type == pygame.MOUSEBUTTONDOWN:

if 50 <= x <= 190 and 700 <= y <= 740:

pygame.quit()

if 230 <= x <= 370 and 700 <= y <= 740:

global over\_pos

over\_pos = []

global mp

mp = np.zeros([15, 15], dtype=int)

run\_game()

if 410 <= x <= 480 and 700 <= y <= 840:

remake()

def check\_win(over\_pos): # 判断五子连心

if len(over\_pos) == 0:

return [0, []]

x = int((over\_pos[-1][0][0]-27)/44)

y = int((over\_pos[-1][0][1]-27)/44) # 标准位置转化成15内坐标

if over\_pos[-1][1] == white\_color:

mp[x][y] = 2 # 表示白子

else:

mp[x][y] = 1 # 表示黑子

a.append([x, y])

# 一列

count = -1

for i in mp:

count += 1

pos1 = []

pos2 = []

qizi = str(i).strip('[]').replace(' ', '')

if '11111' in qizi:

index = qizi.find('11111')

pos1.append([count, index])

pos1.append([count, index + 1])

pos1.append([count, index + 2])

pos1.append([count, index + 3])

pos1.append([count, index + 4])

return [1, pos1]

elif '22222' in qizi:

index = qizi.find('22222')

pos2.append([count, index])

pos2.append([count, index + 1])

pos2.append([count, index + 2])

pos2.append([count, index + 3])

pos2.append([count, index + 4])

return [1, pos1]

# 一行

for j in range(15):

pos1 = []

pos2 = []

for i in range(15):

if mp[i][j] == 1:

pos1.append([i, j])

else:

pos1 = []

if mp[i][j] == 2:

pos2.append([i, j])

else:

pos2 = []

if len(pos1) >= 5:

return [1, pos1]

if len(pos2) >= 5:

return [2, pos2]

# 左上到右下

for i in range(15):

for j in range(15):

pos1 = []

pos2 = []

for k in range(15):

if i+k >= 15 or j+k >= 15:

break

if mp[i+k][j+k] == 1:

pos1.append([i+k, j+k])

else:

pos1 = []

if mp[i+k][j+k] == 2:

pos2.append([i+k, j+k])

else:

pos2 = []

if len(pos1) >= 5:

return [1, pos1]

if len(pos2) >= 5:

return [2, pos2]

# 右上到左下

for i in range(15):

for j in range(15):

pos1 = []

pos2 = []

for k in range(15):

if i+k >= 15 or j-k < 0:

break

if mp[i+k][j-k] == 1:

pos1.append([i+k, j-k])

else:

pos1 = []

if mp[i+k][j-k] == 2:

pos2.append([i+k, j-k])

else:

pos2 = []

if len(pos1) >= 5:

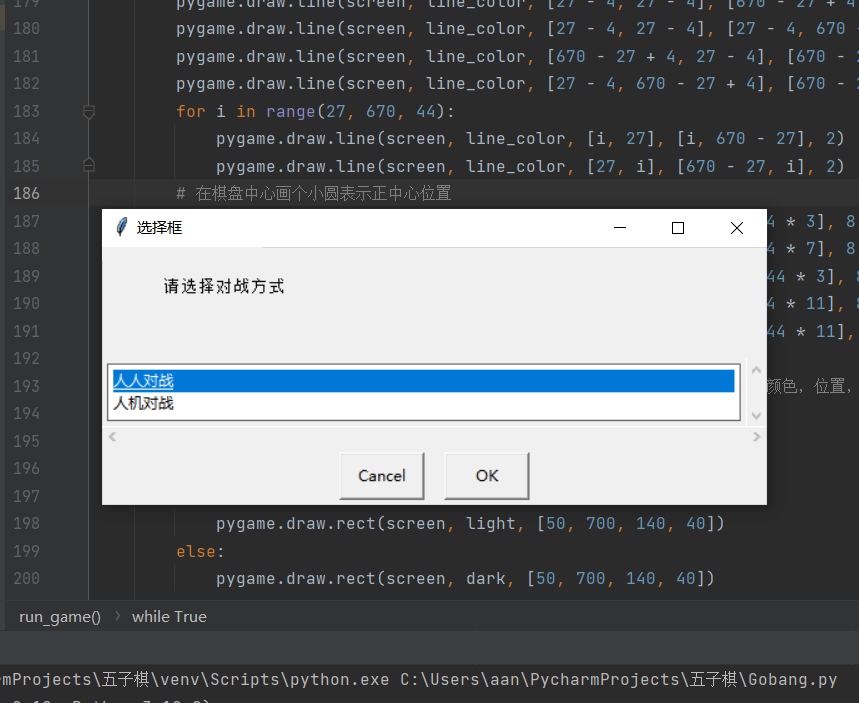
return [1, pos1]

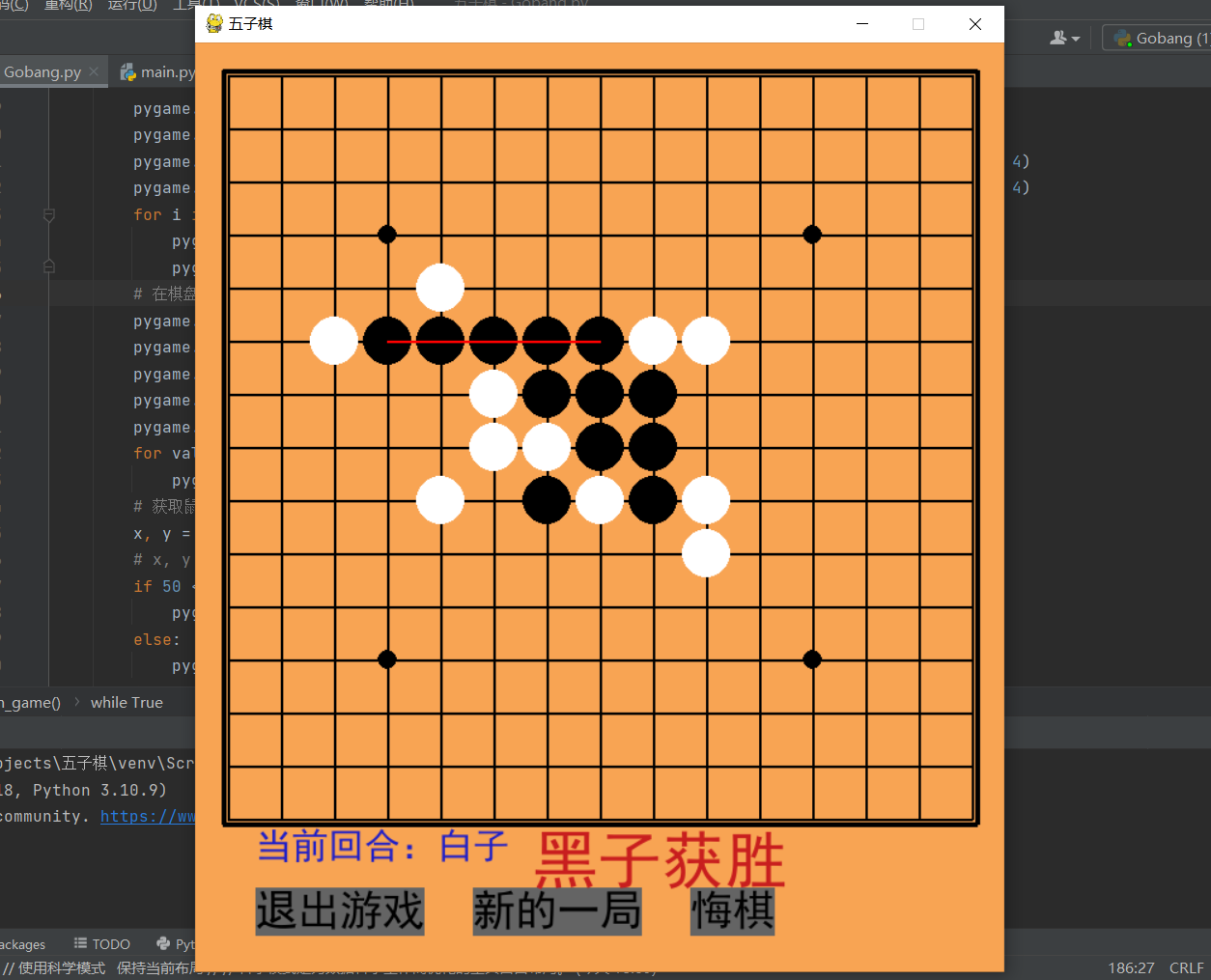
if len(pos2) >= 5:

return [2, pos2]

return [0, []]

**五、项目演示**





**六、心得体会**

学习python并不是一日之功，需要长时间的练习，必须勤于实践，学到一个知识点要善于运用它，代码这种抽象的东西，只有自己动手敲，印象才会深刻，才会记得牢固，python中有许多功能强大的库，要多学习这些库的用法，学会看官方的文档，务必夯实基础，在做一个大项目的时候，要学会分析问题，把一个大项目拆分成许多小任务，逐个完成小任务，提前想好这个项目要怎么做，功能再强大的东西也是最基础的东西组合出来的。